

Synapse Bootcamp - Module 20

Automation in Synapse - Answer Key

Automation in Synapse - Answer Key	1
Answer Key	2
Cron Jobs	2
Exercise 1 Answer	2
Adding Triggers	3
Exercise 2 Answer	3
Trigger Execution	5
Exercise 3 Answer	5

Answer Key

Cron Jobs

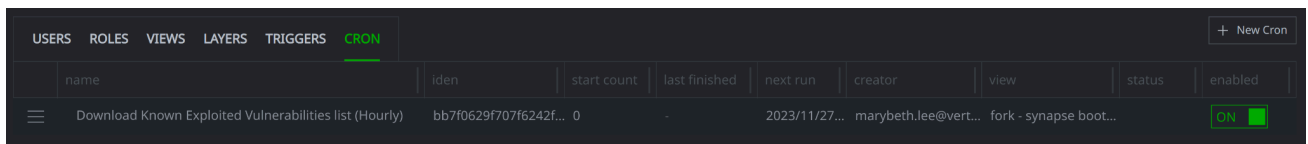
Exercise 1 Answer

Objective:

- Create, manage, and inspect cron jobs.

Question 1: What does your newly added cron job look like?

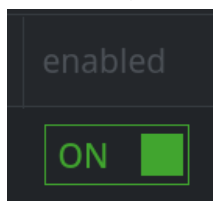
- The job should look similar to the following:



	name	iden	start count	last finished	next run	creator	view	status	enabled
≡	Download Known Exploited Vulnerabilities list (Hourly)	bb7f0629f707f6242f...	0	-	2023/11/27...	marybeth.lee@vert...	fork - synapse boot...	ON	<input checked="" type="checkbox"/>

Question 2: Is the cron job enabled or disabled by default?

- The cron job is **enabled** by default:



Question 3: Are you able to modify your existing cron job to make this change?

- **No**, you cannot make the change.

Once a cron job has been created, you can modify many of the job's properties, but you cannot change the job's schedule. To change the schedule, you need to create a new cron job and disable or delete the old one.

Adding Triggers

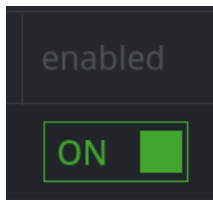
Exercise 2 Answer

Objective:

- Create a trigger to perform basic enrichment of IP addresses as soon as they are created.

Question 1: Is the trigger enabled or disabled by default?

- The trigger is **enabled** by default:



Question 2: What elements of the trigger can be changed after it has been saved?

- Once a trigger is created you can modify:
 - The name and description of the trigger.
 - Whether the trigger runs in the background (asynchronously).
 - The Storm that the trigger executes.

Edit Trigger

Basic IPv4 enrichment

Enrich new IPv4 address nodes with Maxmind and NetTools

Run Trigger in the Background?

ON

1

| maxmind | nettools.dns | nettools.whois

Save

Cancel

You cannot change:

- The condition that fires the trigger
- The form, property, edge, and / or tag the trigger fires on.

To change these elements you must create a new trigger and disable or delete the old one.

Trigger Execution

Exercise 3 Answer

Objective:

- Observe trigger behavior by creating an IPv4 node when the trigger is disabled and when it has been enabled.

Question 1: What properties are present on the new `inet:ipv4` node?

- The IPv4 only has the `:type` property set:

```
inet:ipv4
  8.8.16.1

:type      unicast
.created   2023/11/27 19:14:19.503
```

Question 2: Did your trigger fire?

- **No.** The trigger is **disabled**, so it did not execute when you created the `inet:ipv4` node.

Question 3: What properties appear to be present on the new `inet:ipv4` node?

- Looking at the Details Panel, it appears that only the `:type` property is set:

```
inet:ipv4
  8.8.16.44

:type      unicast
.created   2022/06/13 15:47:42.964
```

Question 4: Did your trigger fire?

- **Yes**, the trigger fired. After re-running your query / re-lifting the node, additional properties are visible:

```
▪ inet:ipv4
  8.8.16.44

▪ :asn      3356
▪ :latlong  37.751,-97.822
▪ :loc      us
▪ :type     unicast
▪ .created  2023/11/27 19:15:57.416
```

Because the trigger runs **asynchronously**, you need to refresh your query to see the properties set when the trigger runs.

When you created the **inet:ipv4** node, Synapse returned the node to you while the trigger was still running **in the background**. This allows you to keep working while the trigger executes. However, you may need to refresh your query to see the changes made by the trigger.

Question 5: What properties are present on the new **inet:ipv4** node?

- The following properties are set on the node:

```
▪ inet:ipv4
  8.8.16.253

▪ :asn      3356
▪ :latlong  37.751,-97.822
▪ :loc      us
▪ :type     unicast
▪ .created  2023/11/27 19:17:21.039
```

When you disable **async** processing, your trigger runs "inline" - as though you added the trigger's Storm commands to the Storm query you used to create the **inet:ipv4** node. You should have noticed a slight delay until your results were displayed - this is because you had to wait for Synapse to create the node **and** run the trigger. However, because Synapse executed everything to completion, your display was refreshed and the new properties were displayed for you.

Whether to run triggers asynchronously is a matter of preference and / or organizational policy:

- Running triggers **in the background** (asynchronously) allows you to keep working, but you may need to refresh your query or return later to see any new data or tags created by the trigger.
- Running triggers **inline** (synchronously) will show you any updates but you will need to wait for the trigger to complete - you are unable to run additional queries, Explore, etc. until the trigger finishes. This may be acceptable for small / fast triggers, but could be problematic for triggers that execute longer Storm queries.

Question 6: Have the additional properties been set on your original IP since you enabled the trigger? Why or why not?

- **No**, the properties have not been set:

```
▪ inet:ipv4
    8.8.16.1

▪ :type      unicast
▪ .created   2023/12/28 01:10:20.528
```

Your trigger is configured to fire on a **node:add** event - when a node is **first** created in Synapse.

Because you created the original **inet:ipv4** while the trigger was **disabled**, the trigger has no effect on the node. Triggers are not "retroactive" - they can only fire on changes that occur **after** the trigger is created and enabled.

The same thing is true for any **inet:ipv4** nodes that existed in Synapse **before** the trigger was **created** - Synapse will not "go back" and run the trigger's query on nodes that were already present.